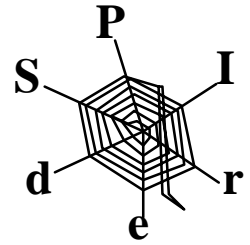


SPIDER— Koerier



Oktober 2005-04

www.st-spider.nl

In Redactioneel

Voor U ligt de vierde SPIDER Koerier van dit jaar. We zijn erg blij met de hoeveelheid kopij en de kwaliteit van de aangeboden artikelen. Twee ervan hebben we deze keer een plek kunnen geven: een stukje over *Defect Injection and Detection* in een virtuele ontwikkelomgeving van Jan van Moll en Jef Jacobs. Daarnaast is er een interessant artikel van de hand van Herman Postema over de resultaten die te behalen zijn met *Architectuur Assessments*. Via oud-bestuurslid Andre Heijstek komen we wat meer te weten over de activiteiten van SEI Europe en blikken we nog even terug naar twee evenementen in de afgelopen periode: de SEPG en onze SPIDER Conferentie. Ook op sponsor-gebied hebben we een verheugende mededeling: PS_Testware is een nieuwe sponsor van SPIDER.

Al met al is de Koerier weer goed gevuld met, naar wij hopen, nuttige en interessante stukjes informatie. Voortaan samengesteld zonder de hulp van Jasper Doornbos. Wij willen via deze weg Jasper bedanken voor zijn inspanningen in de afgelopen jaren voor SPIDER. Mocht U nog een mededeling, suggestie of een artikel hebben waarvan u denkt dat het interessant zou kunnen zijn voor de SPIDER leden, dan laat het gerust weten: koerier@st-spider.nl.

Veel leesplezier gewenst.

Cees Michiels

Inhoudsopgave

In Redactioneel	0
In Van het bestuur	0
In Verslag 8 ^e SPIDER conferentie 2005: Samenwerken in Complexe projecten	2
In Doet het SEI nog iets anders dan CMMI? ..	2
In ESPG conferentie, een impressie van de presentaties vanuit Nederland	3
In Defect injection and detection in virtual development of products	4
In Ps_testware nieuwe sponsor van SPIDER...	8
In Ps_testware als onafhankelijke partner in softwarekwaliteit	8
In Architectuur assessment: wat levert het op?	8
In Kwaliteit beter op de SPIDER 'kaart' geplaatst	12
In De werkgroepen	12
In Oproep nieuwe werkgroep "Personal SPI"	12
In Werkgroep "SPI in kleine organisaties" ..	13
In Werkgroep "Metrieken"	13
In Werkgroep "SPI Invoeringsstrategieën" ..	13

In Werkgroep "Integrale SPI strategieën" ..	13
In Nieuwsberichten & evenementenkalender	13
Evenementen, Conferenties en Seminars	13
Calls for papers	14
In Deelname in SPIDER	14
In Colofon	14

In Van het bestuur

Deze keer maar een kort stukje van mij in de Koerier. De reden: we hebben een volle Koerier, dus er is genoeg te lezen. Uiteraard zijn we heel blij met alle artikelen die we hebben mogen ontvangen, en graag zien we ook weer input voor de volgende Koerier.

De jaarlijkse conferentie is weer achter de rug. Gedurende de dag heb ik al veel goede reacties gekregen van de deelnemers, over de sprekers, locatie en de mogelijkheden om te netwerken. Dat laatste moet zeker niet onderschat worden, het zijn de aanwezige deelnemers, de ambiance, en de kracht van het netwerken die dit een succes maakte. Daarin onderscheid de SPIDER conferentie zich duidelijk van andere evenementen. En, netwerken kan het gehele jaar, via allerlei activiteiten die SPIDER verzorgt.

De activiteiten van SPIDER worden gesponsord door financiële bijdragen van:



Philips.com



Kza.nl



Atosorigin.com



Sogeti.nl



Pstestware.com

Maar, snel door naar de rest van de Koerier. Ik wens jullie veel leesplezier. Zien we elkaar weer op 10 november, de plenaire sessie in Utrecht? Tot dan!

Namens het bestuur,
Ben Linders, voorzitter

n Verslag 8^e SPIder conferentie 2005: Samenwerken in Complexe projecten

Op 22 september was het zover, alweer de 8e jaarlijkse SPIder conferentie. In een nieuwe locatie, de evenementenhal in Gorinchem, nog zo nieuw dat de aanrijroute nog om het complex heen ging. Maar zodra je binnenkwam was het af, en prima verzorgd.

Het was een vol programma, wat begon met een keynote van Gerard Wijers. Hij presenteerde de diverse aanbestedingsmodellen die bedrijven gebruiken voor projecten, al snel werd duidelijk dat met de tegenwoordig hogere eisen die gesteld worden er nieuwe vormen noodzakelijk zijn. Vervolgens presenteerde Noud van Mullekom hoe in het Heathrow Terminal 5 project het procesverbeteringstraject geïntegreerd is met productontwikkeling, waardoor organisaties die verschillen in volwassenheidsniveau effectief samenwerken.

Tijd voor een prima lunch, waarbij de binnentuin uitgebreid benut werd om te netwerken en op die manier het nuttige met het aangename te verenigen. Daarna volgde de parallelsessies, waarin presentaties op het gebied van Product Kwaliteitszorg, Effectieve Samenwerking, en Managen van Verbeteringen aan de orde kwamen. Daarin sprekers uit alle hoeken van de industrie en dienstverlening, en ook vanuit de research: Vrije universiteit en het SEI. Een breed palet van informatie, waaruit de deelnemers haalden wat voor hen bruikbaar is.

Na een korte pauze direct door met het SEI Europe. Geir Fagerhus presenteerde hun visie op software kwaliteit, en de ontwikkelingen die het SEI samen met het bedrijfsleven uitvoert. Daarna een tweeluik over cultuur. Paul Tjia presenteerde aspecten die een rol spelen bij de overbrugging van afstand, tijd en cultuur, daarna liet Shanti Padmanabhan in haar keynote presentatie zien hoe vanuit het design center in India de samenwerking met Europese design center opgezet en verbeterd is.

Diverse sponsors van SPIder waren aanwezig met een stand, deze gelegenheid werd gedurende de gehele dag gebruikt om kennis te maken of om de contacten aan te halen. Alle deelnemers ontvingen een conferentiemap, met daarin een CD met alle presentaties en aanvullend materiaal.

De dag werd afgesloten met een diner, wederom prima verzorgd door de evenementenhal. Een waardige afsluiting van deze goed gevulde conferentiedag.

Graag zien we jullie weer in 2006 op de 9e SPIder jaarconferentie, dus houd de website in de gaten: <http://www.spiderconferentie.nl>

Ben Linders
Dagvoorzitter SPIder conferentie

n Doet het SEI nog iets anders dan CMMI?

In Nederland, en in het bijzonder binnen het SPIder netwerk is het Software Engineering Institute (SEI) goed bekend als beheerder van het CMM en CMMI. Dit betreft echter maar een fractie van wat het SEI doet. In dit artikel, en in volgende artikelen, proberen we een wat completer beeld te schetsen. Eerst maar wat algemene info. Het SEI is in 1984 gestart door het Amerikaanse ministerie van defensie (Department of Defense, DoD) om de 'state-of-the-practice' van software engineering te verbeteren. Het SEI is onderdeel van de prestigieuze Carnegie Mellon University in Pittsburgh. Recent heeft het SEI opnieuw een contract voor 5 jaar afgesloten met het DoD voor 411 miljoen dollar. Dit betreft ongeveer 20% van het SEI budget, verder komt de financiering voornamelijk uit gezamenlijke onderzoeksprojecten met industrie en defensie, en deels uit licenties, trainingen, etc.

De primaire missie van het SEI is om software engineering te veranderen van ad-hoc en arbeidsintensief naar een beheerst en door technologie ondersteund vakgebied.

Hoewel het SEI een Amerikaans instituut is en de toezichtsraad gedomineerd wordt door het DoD, worden de SEI technologieën wereldwijd toegepast, zowel binnen defensie als binnen de industrie en de IT. Vanwege de grote vraag naar meer ondersteuning binnen Europa werd in 2003 in Frankfurt een Europees kantoor van het SEI geopend. Van hieruit worden alle SEI technologieën ondersteund. SEI-Europa streeft ernaar mensen in dienst te nemen die alle (belangrijke) Europese talen spreken en breed in Europa actief te zijn. Van de huidige 10 werknemers is er een sterk Nederlands-sprekend contingent: Hans Sassenburg is binnen SPIder goed bekend als medeoprichter en eerste voorzitter; Peter Leeson, hoewel Brits staatsburger spreekt hij uitstekend Vlaams; en André Heijstek, op dit moment de enige SEI medewerker die ook in Nederland woont. De andere medewerkers hebben de Zweedse/Noorse, Poolse, Ierse en Duitse nationaliteit.

Maar, dit artikel zou gaan over wat het SEI doet, dus hieronder beschrijven we kort de zogenaamde SEI Focus Areas. In volgende artikelen zullen we deze focus areas nader beschrijven.

Capability Maturity Model Integration (CMMI)

Omdat dit programma waarschijnlijk al goed bekend is zullen we het hier niet verder beschrijven.

Integration of Software-Intensive Systems (ISIS)

Meer en meer systemen kun je op dit moment beschrijven als systemen-van-systemen, opgebouwd uit maatwerk en pakketten of commercial-off-the-shelf-software (COTS). Zeker stellen dat al deze (deel)systemen goed met elkaar samenwerken, ook bij wijzigingen en uitbreidingen die vooraf niet voorzien werden is bepaald geen triviale taak. Het ISIS team doet onderzoek naar deze klassen van systemen en biedt methodes en tools aan om met deze uitdagingen om te gaan.

Performance-Critical Systems (PCS)

In dit initiatief voert het SEI onderzoek uit om de voorspelbaarheid en performance van real-time systemen te verbeteren. Het bekendste resultaat van deze groep is de Architecture Analysis and Design Language (AADL), een taal om real-time systemen te specificeren. Deze taal is door de Society of Automotive Engineers inmiddels als standaard binnen haar domein benoemd, en wordt verder nog veel toegepast in de ruimtevaart. Maar er is geen enkele reden om de toepassing te beperken tot deze domeinen.

Product Line Practice (PLP)

Ook de PLP groep streeft naar re-use, maar dan op een andere manier. Veel producten behoren tot een productfamilie. De producten zelf worden op een standaard platform gebouwd en klant- of marktspecifieke varianten worden daarvan afgeleid. De PLP-groep zorgt ervoor dat dit type ontwikkeling systematisch kan worden aangepakt.

Software Architecture Technology (SAT)

In dit initiatief biedt het SEI effectieve methoden aan om systemen onder architectuur te ontwikkelen. De Architecture Tradeoff Analysis Methodsm (ATAMsm) biedt de mogelijkheid om een architectuur systematisch te analyseren en de gewenste kwaliteitsattributen in kaart te brengen en tegen elkaar af te wegen. Diverse populaire boeken zijn door deze groep gepubliceerd: "Software Architecture in Practice", "Documenting Software Architectures – Views and Beyond", "Evaluating Software Architectures: Methods and Case Studies". Centraal in al deze boeken staat het idee dat architecturen ontworpen moeten worden met een specifiek doel – de kwaliteitsattributen.

Software Engineering Measurement and Analysis (SEMA)

Het meest zichtbare werk van de SEMA groep is de 6-maandelijkse publicatie van het process maturity profile met gegevens over CMM en CMMI training, assessments, etc. <http://www.sei.cmu.edu/appraisal-program/profile/about.html>. Maar SEMA doet veel meer. Deze groep herbergt de SEI specialisten op het gebied van metingen en statistische analyse. Hier wordt ook veel onderzoek gedaan naar de gecombineerde toepassing van CMMI en Six Sigma. SEMA biedt diverse uitstekende cursussen

aan op het gebied van measurements – deels worden deze cursussen ook in Europa aangeboden

Network Systems Survivability (NSS)

Het doel van NSS is om technologie en methodes te ontwikkelen om te zorgen dat systemen bestand zijn tegen aanvallen van hackers, virussen, etc. of dat systemen ondanks succesvolle aanvallen toch nog een groot deel van hun diensten kunnen blijven leveren. NSS heeft in 1988 het CERT Coordination Center opgezet (Computer Emergency Response Team) dat continu inventariseert welke duistere praktijken op internet plaatsvinden. Het CERT concept is inmiddels wereldwijd verspreid, ook Nederland heeft er enkele: www.cert.nl.

NSS biedt diverse cursussen aan op het gebied van software security en voor het opzetten van CERTs, deels ook in Frankfurt.

Team Software Process (TSP)

Watts Humphrey, de vader van het CMM, had verwacht dat als organisaties op level 3 of hoger zouden komen, dat dan de ontwikkelaars ook hun methodes voor het dagelijkse ontwikkelwerk zouden verbeteren. Dit gebeurde echter niet. Organisaties oogstten wel voordelen van hogere maturity levels, maar nog niet in de mate die Watts had verwacht. Als antwoord hierop heeft Watts 2 nieuwe modellen ontwikkeld: het Personal Software Process (PSP) en het Team Software Process (TSP).

Voor al deze focus areas past het SEI haar standaard business proces toe: *create*, *apply* en *amplify*. In de *create* fase ontwikkelt het SEI nieuwe technologieën en methoden, vaak in samenwerking met universiteiten en partners uit de industrie of defensie. In de *apply* fase wordt deze nieuwe technologie als pilot toegepast in de praktijk. Wanneer een technologie volwassen is geworden wordt deze in de *amplify* fase doorgegeven aan de SEI-partners die er dan voor zorgen dat zoveel mogelijk bedrijven gebruik kunnen maken van de nieuwe methoden.

SEI-Europa is actief binnen alle 3 de fasen van het SEI business proces en is geïnteresseerd in samenwerking met Europese bedrijven. Heeft u interesse in samenwerking of vragen over dit artikel, schroom dan niet om contact met ons op te nemen.

Andre Heijstek, andreh@sei.cmu.edu

Hans Sassenburg, hanss@sei.cmu.edu

Peter Leeson, pleeson@sei.cmu.edu

Website: <http://www.sei.cmu.edu/about/europe/>

n ESPG conferentie, een impressie van de presentaties vanuit Nederland

Op de European Software Engineering Process Group conferentie in Londen (www.espi.org) hebben een aantal Nederlandse sprekers een presentatie gegeven. Hieronder een korte impressie.

Hans Sassenburg presenteerde de resultaten van zijn promotieonderzoek met als onderwerp het vrijgaveproces van software applicaties. Op basis

van een theoretisch onderzoek gecombineerd met de resultaten van 7 case studies is een methodologie ontworpen. Deze methodologie combineert een economisch perspectief met een besluitvormings perspectief en stelt een organisatie in staat haar vrijgaveprocessen te begrijpen, te beoordelen en dus ook te verbeteren. Validatie van de eigenschappen van de methodologie heeft in nog eens 3 case studies plaatsgevonden. Hans hoopt dit jaar te promoveren op zijn onderzoek aan de Rijksuniversiteit Groningen.

In een gecombineerde presentatie werd de invoering van Measurement & Analysis bij Bosch Engineering vanuit 3 gezichtspunten bekeken: de interne SEPG leider, de consultant en de appraiser. Bosch Engineering heeft een measurement-programma opgezet door in brainstorm sessies met measurement customers de information needs te bepalen, en uit een standaard measurement kit (ontwikkeld door Q-Labs) de meest geschikte measurements gekozen en enkele nieuwe ontwikkeld. De projecten gebruiken een 2-wekelijks bijgewerkte dashboard chart die bestaat uit 9 indicatoren. Elke indicator krijgt op basis van vaste criteria automatisch een rood/geel/groene rand om de status aan te geven. De appraiser (Andre Heijstek) deed een mapping van de gevonden practices naar het CMMI model en besprak de institutionalization op basis van methoden uit de gezinsplanning.

Frans Maas presenteerde de voordelen van de Critical Chain project management methode. Waar projectmanagers zich traditioneel richten op 'taak op tijd af', verlegt Critical Chain de focus volledig naar 'project op tijd af'. In het kritieke pad en de aanvoerlijnen daar naartoe bouwt deze methode buffers in, die de gevolgen van onzekerheden in taakdoorlooptijden absorberen en verminderen. Dit verkort de geplande projectduur en zorgt ervoor dat projecten beter aan de gestelde doelen voldoen. Een zelfde presentatie heeft Frans gegeven op de op de plenaire sessie van 29 november 2004, ook is in de Koeriers van mei en juni een artikel van hem gepubliceerd.

Door Robert van Lieshout is een presentatie gegeven over formele methoden. Deze methoden winnen de laatste tijd weer aan populariteit. Een voorbeeld daarvan is I-Mathic. I-Mathic is een formele software ontwikkelmethode die elementen uit Cleanroom Software Engineering integreert met elementen uit andere methoden en voorziet van toolondersteuning. Eisen worden omgezet in een formele specificatie, waaruit state handling code, test scripts, en een CSP model worden gegenereerd. Het CSP model wordt gevalideerd met een COTS model checking tool om te kunnen bewijzen dat de specificatie correct is. Aangezien de code gegenereerd wordt vanuit deze specificatie is de kans op fouten daarin zeer klein. I-Mathic is toegepast bij klanten van Imtech met kostenbesparende resultaten: hoge productiviteit (meer dan 120 elocs per mandag) en weinig post-release defects (minder dan 0.3 per 1000 elocs). De methode is ook met succes toegepast in bestaande software om de oorzaak van moeilijk te reproduceren foutsituaties op te sporen.

Nederland was goed vertegenwoordigd op de ESEPG, met zowel presentaties vanuit de onderzoekswereld als uit de praktijk. Ook waren er aardig wat deelnemers uit Nederland. Alles bij elkaar een prima conferentie, die we namens SPIder kunnen aanbevelen. In 2006 zal de conferentie wederom in juni plaatsvinden. Meer info: www.espi.org.

Ben Linders, voorzitter

n Defect injection and detection in virtual development of products

Introduction

Industry wide, an increasing trend towards the development of products by virtual teams is observed. Virtual development, also referred to as global development, is the development of products by teams distributed across space, time, and organization boundaries, linked by webs of interactive technology.

Contemporary and frequently cited publications on virtual development, Carmel, and Karolak use the term "global software development" [Carmel,1998; Karolak, 1999]. However, we prefer the term "virtual development", because the scope of development is not necessarily restricted to software, and because the development team need not necessarily be scattered around the globe. Being in another building or on a different floor of the same building, or even at the other end of a long corridor, can be sufficient to label it as virtual development.

Virtual development of products is much more complex than even the most complex project managed entirely in house. Carmel recognizes the following three unique aspects in which virtual development differs from all in-house development:

- *Distance* between development sites has a direct impact on project control, coordination and communication
- *Time zone* differences between development sites make it even harder to communicate, impacting project control and coordination
- *Cultural differences* between development sites may lead to mistrust, miscommunication and lack of cohesion.

On the basis of these findings, Carmel identified five problem areas that act as "centrifugal forces, driving the global development team apart". We interpret Carmel's statement of problem areas that are "driving the global development team apart" as problem areas that potentially threaten the delivery of the product in time, within budget and with the specified or implicitly expected product quality. These problem areas are: (1) geographic dispersion, (2) control and coordination breakdown, (3) loss of communication richness, (4) loss of "teamness" and (5) cultural differences.

Carmel, Karolak and many others are focused on the managerial and collaboration aspects of the organization and execution of virtual development projects. The emphasis is on timely delivery of the product within budget. The effects of virtual development on product quality are not or only

marginally addressed despite strong indications of adverse effects of product quality. The effects on product quality are important because lack of product quality will keep burdening the organization far beyond the termination of the development project; e.g., with high and prolonged maintenance costs, loss of market share, lawsuits, and costly product recalls.

One can look at quality from a variety of different viewpoints, each being relevant to different stakeholders: the transcendental view, the manufacturing view, the product view, the user's view, and the value-for-money view. We consider the manufacturing view, usually encapsulated in the phrase "conformance to specification", as our base view on product quality. Non-conformances to specifications (which we shall call defects) will typically have a negative impact on product quality whatever the point of view.

Given (1) the increasing trend towards virtual product development, (2) the significant risk of adverse effects of virtual development on product quality, (3) the observation that the effects of virtual development on product quality are only marginally addressed in literature, and (4) the importance of product quality, we have initiated a research project with the ultimate goal being the design of a framework for reducing the number of residual defects in products developed by virtual teams.

Virtual development projects are typically organized as a system project and related sub-projects. Consequently, defect injection becomes dispersed over multiple sub-projects. Likewise, defect detection becomes dispersed. Causes for defect injection and non- or late detection have different locations and different assignment of causes. Therefore, and because product creation on the one hand, and product verification and validation on the other hand are engineering activities of a different nature, our research project is directed towards two lines of investigation: (1) minimizing defect injection and (2) maximizing defect detection and removal.

Problem types in virtual development

An investigation of a number of past projects was carried out to learn more about the origin of at least the most severe problem types in virtual development environments [van Moll et al, 2004]. Severity is interpreted as having a significant negative impact on schedule, cost or product quality. The investigation included 22 projects, representative for virtual product development, from companies in the Netherlands, Germany and Belgium.

Of all severe problems, about 65% was directly related to the application of practices often encountered in virtual (and software-intensive) product development. These practices include **reuse of existing components**, **application of third-party components** and **development by external organizations**. Some projects applied a combination of practices. Figure 1 shows an overview of the practices and the number of projects that reported severe problems in applying them. For each practice, the actual problems experienced, are briefly summarized below.

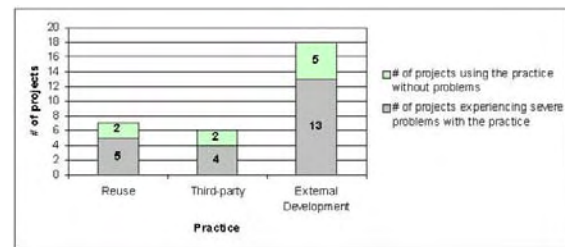


Figure 1. Overview of projects reporting severe problems regarding defect detection per practice

Reuse of existing components

Reuse is a way to reduce total development effort or to minimize product quality risks by using an already existing and tested component developed earlier in the same organization. Commonly observed problems were:

- Omitting the scheduling of phases or activities to cover the effort needed for proper reuse of the component.
- Not planning any effort for determining whether the test conditions originally used during testing of the component are still valid in the target environment of the current application.

Application of third-party components

Third-party components are existing components that are supplied by organizations commercially exploiting them. Using third-party components is a form of reuse. The main difference being that the provider of the component is located outside the own company. Commonly observed problems were:

- Not planning any effort needed to customize or tailor the component to suit specific applications.
- The performance of an intake-test is rarely practiced; system projects were often confronted with problems while integrating the component.

Development by external organizations

Today it is common practice to leave the development of components to separate development teams within the organization or to external subcontractors. Whatever their organizational relationships with the overall system project, these teams always need input from the system project. Likewise, there is always a moment in time where they deliver products back to the system project. Commonly observed problems were:

- Any problems related to the component appear only at the time of, or just before, acceptance of the component by the system project
- The test strategy for the component did not match with the strategy defined for the overall system, causing a too low overall test coverage
- Defects in product parts not addressed were allowed to propagate until stages of the project where their late detection caused unacceptably high project pressure.

Causes of defect injection and non- or late-detection

Insight in the problem types is a first step towards our ultimate goal, the design of a framework for reducing the number of residual defects in products developed by virtual teams. A next step is an

exploration of causes of defect injection and causes of non- or late- defect detection, typical of virtual development [Jacobs et al, 2005]. Such knowledge is required to eventually devise improvement measures.

In the first instance, we wanted to investigate in-depth the causes of defects in a number of virtual development projects. However, such an approach turned out to be unfeasible for a variety of reasons. We therefore applied an approach based upon accumulated expert knowledge concerning defect causes in virtual development projects.

We were particularly interested in (1) typical causes for the *injection* of defects in virtual development, and (2) typical causes for *non- or late-detection* of defects in virtual development. Non-detection of defects refers to defects that are not detected during the development and verification and validation phases of a project, but are discovered after release in the field. Late-detection of defects refers to defects that were found during the development and verification and validation phases, but should have been detected in earlier phases.

Six analysis meetings, with selected experts as participants, were conducted to identify causes for defect injection and non- or late-detection. The meetings very much resembled the causal analysis meetings as seen in a Defect Causal Analysis (DCA) process. In conventional DCA, systematic errors are identified based on quantitative data from a project's problem database. In our approach we aimed at identifying defect types and associated causes based on qualitative information mined from the participating experts' experiences.

In each meeting, a lifecycle-centric view on product creation was taken, meaning that the individual development and test phases were considered and discussed separately in the order they appear in a generic V-lifecycle for virtual development projects, shown in figure 2.

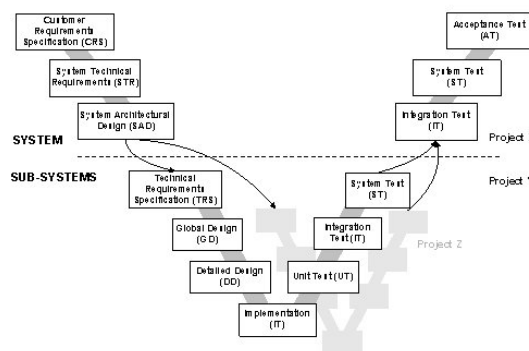


Figure 2. Generic V-lifecycle as deployed in a typical virtual development context

Identification of causes started by brainstorming about defects in the risk areas of Communication, Process, Organization and Technology. By using a cause-effect graphing technique, the participants tried to systematically identify all possible causes for defect injection and non- or late defect detection. Only those causes that typically relate to virtual development were taken into account: defect causes that are equally likely in both virtual development as in all in-house development were left out.

Below we summarize the main findings:

Injection of defects

- The transitions between lifecycles of sub-projects from virtual development projects are particularly defect sensitive.
- In virtual development, *new types* of defects (i.e. exclusively occurring in virtual development context) are not to be expected. Rather, virtual development (1) introduces *new causes* for known defect types and (2) increases the likelihood of defect causes existing in all-in-house development. Consequently, dispersion of defects as well as an increased likelihood of injection lead to a higher number of defects in the delivered product.
- Defect injection is significantly increased in situations where changes in the requirements, design or implementation are being handled. While non-adequate handling of changes is already severe in all-in-house development, virtual development projects even seem to aggravate the effects, resulting in additional defect injection causes.
- Communication is a main risk category for the injection of defects in the Requirements Specification phase. This is plausible: the nature of activities on the work products in this phase imply intensive information exchange between multiple groups or persons involved.
- A concentration of defect causes occurs in the Integration Test phase at the system level. Most causes were assigned to the risk categories Process and Organization. This is plausible because the nature of the activities in this phase cross organizational boundaries where organizational style, structure and processes (like testing) can hinder effective component integration and problem solving.

Non-detection or late-detection of defects

- Causes of non- or late-detection are concentrated in the Integration Test phase of the system-level project (project X, see figure 2). At this lifecycle transition, the components developed in the sub-projects are integrated and subsequently tested. Dispersed defects (especially from the Requirements phase transitions) will become painfully visible here.
- Lacking or insufficient test coordination (over the entire virtual development project) seems to be the major cause of non or late-detection of defects. Project delays threaten the execution of a proper integration test. Projects tend to rely upon the subsequent system test as a fallback, not or insufficiently realizing that certain types of integration related defects cannot be detected at this later phase. Even in in-house projects, insufficient attention for test coordination and test approach by project management often results in problems with product quality. In virtual development projects the effects of lacking or insufficient test coordination seem to be aggravated resulting in additional causes for non- or late-detection.
- Risk categories Technology and Process show the highest number of causes in the Integration Test phase. This is plausible, as testing-related

problems mainly stem from technical aspects like test environments, testing tools and processes like test management and change control.

- System Test shows a similar cause pattern. This might be explained by the fact that system testing typically finds different types of defects than integration testing. And although different types of defects are to be found in integration and system test, both have similar causes for non- or late detection.
- Causes of non- or late defect detection are also found in the Requirements phase. These non- or late detection causes exist in addition to the already present concentration of injection causes in that same phase. That injection and non- or late detection causes coincide in the same phase is worrying as this might increase the propagation and dispersion of defects into sub-projects: the Requirements phase has potential causes of defect injection and at the same time prevents these causes from being detected.

Possible effect of improvement measures

One of the most important findings so far is that the transitions between the lifecycles of the individual projects that together form the virtual development project are particularly error prone. As lifecycle transitions are typical of virtual development projects, we wanted to assess the effects of improvement measures, specifically geared to the many lifecycle transitions that occur in a regular virtual development project.

The main improvement measure is to explicitly model the overall project lifecycle, resulting in a clear map of the project, in which all transitions between the lifecycles of individual subprojects are clearly visible. This measure exposes defect-sensitive areas and enables the appropriate development and test phases to be modeled into the overall project lifecycle. It allows for a timely application of appropriate defect detection measures (e.g. reviews, intake inspections of specifications and test object, dynamic testing) at the right moment by the right parties during the entire development process, particularly at lifecycle transitions. By appropriate modeling of the overall project lifecycle and applying adequate defect detection measures at the transitions of the related lifecycles, the occurrence of defects can be prevented and fewer defects can be expected. To get at least an indication of the influences of proper lifecycle modeling an in-depth post-mortem investigation of a real-life project was carried out.

The selected project was to develop a system intended for the professional/industrial market; it comprised hardware, software and firmware components that were developed by separate development teams. An overall system project had been defined that was responsible for respectively the development and testing of the main application and system software, and the integration of the sub-components.

Some of the components were contracted out to companies who specialized in the development of these types of components. Other components were created by reusing and extending existing

components. Each component project had its own project management plan. Their project managers reported to the overall system project manager.

For developing the system, an incremental development approach was chosen. Actual planning was done at the overall system level.

Its overall lifecycle was not consciously modeled, and as a result, the error prone transitions were not recognized. Not surprisingly, it suffered from many of the typical virtual development related problems as outlined previously in this article. Consequently, many problems reports were received.

As a first step, the overall lifecycle was remodeled, based on the same information as was known at the time of initiating the project. This *remodeled* lifecycle can then be used as a reference to compare with the *actual* development lifecycle that has been used in the project. Secondly we "added" appropriate measures at the transitions, for example intake of requirement specifications and test objects, review of test reports, and an integration test strategy.

Next, an analysis team was formed, consisting of engineers and architects who were actually involved in the project, and who had detailed inside knowledge about the project and the defects reported. For each defect reported, this team estimated the probability that the defect would a) not have been injected or b) would have been detected earlier, when the remodeled lifecycle would have been used in combination with the measures at the transitions of the related lifecycles.

In total 143 problem reports existed in the project's problem database. The analysis showed that 43% of the problems reported was caused by defects in the components developed by external parties. In the components' development, the number of interpretation errors was more than twice as high as the number of implementation errors (42 versus 20). The majority of the interpretation errors was caused by component specifications which were not reviewed with the participation of system architects. While estimating the probability of prevention, it became clear to the analysis team what could have been gained: adequate lifecycle modeling and additional measures at lifecycle transitions could have prevented 42% of the total number of problem reports from occurring in the system test. As directly product-related problem reports are concerned, 32% could have been prevented. The remaining 10% of the problem reports was caused by erroneous test specifications.

Conclusions

Virtual product development can be viewed as a hierarchy of distributed and concurrent projects. Because of the distributed nature, lifecycle transitions between related subprojects are inherent to virtual product development. A case study involving 22 projects indicated that these transitions are particularly error prone. An analysis of causes of defect injection and non- or late- defect detection confirmed that the transitions are error prone and provides input for devising improvement measures. A post-mortem analysis of a virtual development project suggests that by focusing the attention of improvement measures on transitions, a significant

amount of defects (32%) could have been prevented.

References

Carmel E *Global Software Teams: Collaborating Across Borders and Time Zones*. Prentice Hall, 1998.

Jacobs JC, Moll JH. van, Krause P, Kusters RJ, Trienekens JJM, Brombacher AC Exploring Defect Causes in Products Developed by Virtual Teams. *Information and Software Technology* 2005; 47(6): 399-410.

Karolak DW *Global Software Development*. IEEE CS, Los Alamitos, USA, 1999.

Moll JH van, Jacobs JC, Kusters RJ, Trienekens JJM Defect Detection Oriented Lifecycle Modeling in Complex Product Development. *Information and Software Technology* 2004; 46(10):665-675.

About the authors

Jan van Moll (jan.van.moll@sioux.nl) studied *Technical Physics and is working as Process Expert at Sioux Embedded Systems BV, Eindhoven, the Netherlands, and is specialized in verification and validation*.

Jef Jacobs (jef.jacobs@philips.com) studied *Electrical Engineering and Statistics. He is with Philips Semiconductors, in Eindhoven, The Netherlands, where he is globally responsible for business and development processes*.

They are currently conducting a joint PhD research at the Eindhoven University of Technology, faculty of Technology Management, studying the effect of virtual development on product quality.

n Ps_testware nieuwe sponsor van SPIder

Sinds 1 augustus heeft SPIder er een nieuwe sponsor bij: ps_testware. Diegenen die op de conferentie zijn geweest hebben al kennis kunnen maken. Hieronder stelt ps_testware zichzelf voor. Uiteraard is SPIder zeer blij met deze sponsoring, hij maakt het mede mogelijk om onze activiteiten op een professionele manier te doen voor jullie, de donateurs en leden van SPIder!

Ben Linders, voorzitter

n Ps_testware als onafhankelijke partner in softwarekwaliteit

Problemen met de kwaliteit van je software? Hulp nodig bij het structureren van je requirements of testactiviteiten? Inzicht nodig in wat gestructureerd testen en validatie voor je kan betekenen en hoe je dit implementeert? ps_testware kan hiervoor een oplossing bieden.



ps_testware is een Belgisch/Nederlands consultancy bedrijf een met kantoor in Gorinchem voor de Nederlandse markt en is al meer dan 10 jaar gespecialiseerd in softwarekwaliteit

(gestructureerd software testing en computer system validation). Sinds kort is ps_testware sponsor van de SPIder en hierbij willen we onze duit in het zakje van Software Process Improvement doen, omdat we vanuit onze invalshoek zien dat ook daar verbetering nog mogelijk is.

Een 60-tal consultants van ps_testware werkt in Nederland, België en Frankrijk aan de verbetering van softwarekwaliteit bij grote klanten (vooral bank-, verzekerings-, farmaceutische en energiesector). Deze medewerkers zijn stuk voor stuk ISEB gecertificeerd en passen een bewezen methodologie toe (gebaseerd op het I-Model), waarbij de diverse fases in het test- en validatieproces op een gestructureerde wijze worden doorlopen om zo tot een kwaliteitsvol softwareproduct te komen en in één klap ook kostbare tijd te winnen.

Een belangrijk aspect van onze methodologie is het werken met een URH (User Requirements Hierarchy). Hierbij wordt het mogelijk de gebruikers- en testvereisten op een eenvoudige manier te beheren en als basis te gebruiken voor zowel de bouwfase als testontwikkeling en –uitvoering. De methode die hiervoor wordt gebruikt, noemen we RUM™ (Risk-Assessed User Requirements Management).

ps_testware is enkel tevreden als de klant tevreden is. Wat de klant wenst, is ook wat wij willen en zo komen we tot een gemeenschappelijk doel: het opleveren van een kwaliteitsvol product volgens een logisch gestructureerd en kwaliteitsvol proces. ps_testware levert test- en validatiediensten onder diverse vorm: consultancy, coaching, training, coördinatie, management en outsourcing. Onze methodologie passen we zowel toe op manuele testprocessen als testautomatisering (regressietesten en performantietesten) en we testen allerhande software (administratief, embedded, logistiek..).

ps_testware durft slecht nieuws te brengen als dit nodig is. Advocaat van de duivel kunnen we enkel zijn omdat we optreden als onafhankelijke (test)partner om kwaliteit op een hoger niveau te brengen.

Contactpersoon: Alain Bultink,
Alain.Bultink@pstestware.com

n Architectuur assessment: wat levert het op?

Het belang van kwalitatief hoogwaardige architecturen voor software intensieve systemen neemt in hoog tempo toe. Dit heeft geleid tot een sterke behoefte aan methodieken om het kwaliteitsniveau van een architectuur te beoordelen. Herman Postema bericht over zijn ervaringen met het uitvoeren van deze zogeheten architectuur assessments.

Belang van architectuur

In de wereld van de software intensieve systemen is de afgelopen jaren een sterk besef gegroeid dat de systeemarchitectuur een belangrijke bijdrage levert

aan het realiseren van bedrijfsdoelstellingen. IEEE 1471 definieert architectuur als *de fundamentele organisatie van het systeem in componenten, de relaties van deze componenten met elkaar en met hun omgeving, en de principes die richting geven aan haar ontwerp en evolutie*. De architectuur blijkt in grote mate bepalend te zijn voor de zogenaamde niet-functionele eigenschappen van het product. Voorbeelden hiervan zijn de mate van betrouwbaarheid, beschikbaarheid, onderhoudbaarheid, schaalbaarheid, herbruikbaarheid, resourceverbruik en veiligheid. Vaak vertegenwoordigen deze eigenschappen essentiële *business drivers* van een productontwikkelaar. Een voorbeeld van zo'n *key business driver* is de *Field Call Rate* zoals Philips die hanteert als de mate waarin het product (niet) beantwoordt aan klantverwachtingen en dus klachten oplevert voor bijvoorbeeld een televisietoestel. Maar ook zaken als innovatiekracht, *cost of ownership* en *time-to-market* zijn veel voorkomende *key business drivers*. Hoewel dit geen niet-functionele producteigenschappen zijn worden ze er wel sterk door beïnvloed, en daarmee dus ook door de systeemarchitectuur.

Kwaliteit van architecturen

Business drivers varen dus wel bij kwalitatief hoogwaardige systeemarchitecturen. Zo'n architectuur is het resultaat van een groot aantal technische keuzes dat het team van architecten heeft gemaakt. De kwaliteit van de architectuur is hoog als deze keuzes maximaal ondersteunend zijn aan de niet-functionele eisen die aan het product worden gesteld. Uiteraard moeten deze keuzes dan ook nog vallen binnen een aantal randvoorwaarden. Deze bakenen in feite de oplossingsruimte af die de architect tot zijn beschikking heeft. Voorbeelden van randvoorwaarden zijn applicatiestandaarden, standaard technologiekeuzes, interface afspraken en het gebruik van externe componenten. Maar ook randvoorwaarden die niet product-gerelateerd zijn kunnen de architecten sterk in hun keuzevrijheid beperken. Denk hierbij aan het beschikbare ontwikkelbudget, de gekozen ontwerp-methodiek en de doorlooptijd die het ontwikkelteam ter beschikking staat.

Architectuur assessment

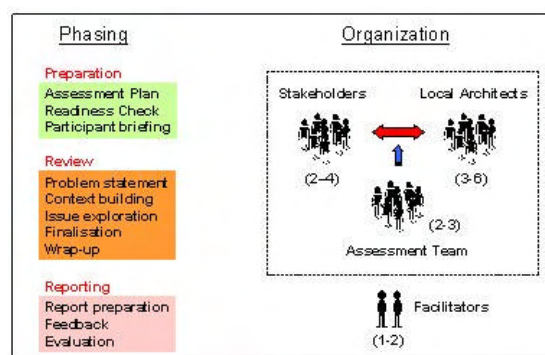
Door het toenemende besef dat kwalitatief hoogwaardige architecturen een voorwaarde voor zakelijk succes zijn, is er de afgelopen jaren veel te doen geweest rond de vraag hoe die kwaliteit op een betrouwbare wijze kan worden vastgesteld. Onder de noemer architectuur assessment zijn hiertoe inmiddels vanuit diverse hoeken methodieken ontwikkeld. Algemeen bekend is de *Architecture Trade-off Analysis Method (ATAM)* van het Amerikaanse Software Engineering Institute (SEI). Daarnaast hebben enkele grote bedrijven, waaronder AT&T en Nokia, al enkele jaren geleden hun eigen methodiek ontwikkeld. Tot slot worden ICT dienstverleners in toenemende mate gevraagd om architectuur assessments voor hun klanten uit te voeren. Zij spelen hier op in door het ontwikkelen van eigen methodieken die zo goed mogelijk aansluiten bij de behoeften van de klant. Zo hanteert LogicaCMG voor embedded-softwareleveranciers een methodiek die onder meer is gebaseerd op de resultaten van de internationale

Software Architecture Review and Assessment (SARA) werkgroep. Deze werkgroep is enkele jaren actief geweest. Ze was samengesteld uit vertegenwoordigers van grote (embedded-)softwareleveranciers, waaronder Philips, Siemens, Nokia, IBM en Rational. Het doel van de werkgroep was om de ervaringen en "*best practices*" op het gebied van architectuur assessment te inventariseren en vast te leggen.

Globale aanpak

Een architectuur assessment uitgevoerd door LogicaCMG is een systematische analyse van een systeem- of softwarearchitectuur. Het doel is om het kwaliteitsniveau van de architectuur vast te stellen en aanbevelingen voor verbetering te geven. Tijdens het assessment worden de technische architectuurkeuzes systematisch geconfronteerd met de niet-functionele eisen die aan het product worden gesteld en met de randvoorwaarden. De architectuurkeuzes zijn gemaakt door het team van architecten dat aan het product werkt. De eisen en randvoorwaarden zijn afkomstig van de zogenaamde *stakeholders* (belanghebbenden) zoals marketing, product-management, ontwikkeling, onderhoud, kwaliteit, service en productie.

In de dagelijkse praktijk blijkt de communicatie tussen *stakeholders* en architecten vaak verre van optimaal te zijn. Om die reden vindt het assessment plaats in de vorm van een plenaire workshop waar beide groepen actief met elkaar worden geconfronteerd (Figuur 1). Deze workshop kan 1 tot 3 dagen in beslag nemen, waarbij tussen de 8 en 15 personen aanwezig zijn. Als eerste stap wordt toegewerkt naar een gemeenschappelijk begrip van de relevante *key business drivers*, niet-functionele eisen, randvoorwaarden en architectuurkeuzes. Op basis van deze gemeenschappelijke context worden de architectuur-zorgen in kaart gebracht die leven bij de *stakeholders* en de architecten. Het grote deel van de workshop bestaat vervolgens uit het in detail onderzoeken van deze potentiële probleemgebieden. Hiervoor staan verschillende technieken ter beschikking die uiteindelijk resulteren in uitspraken over de kwaliteit van de architectuurkeuzes. Als laatste stap worden conclusies en aanbevelingen uitgewerkt.



Figuur 1: Globale aanpak en organisatie van een architectuur assessment

Tijdens de workshop is een assessment-team aanwezig dat bestaat uit ervaren architecten die niet

bij de ontwikkeling van het product betrokken zijn geweest. LogicaCMG kan hiertoe putten uit een intern en extern netwerk van architecten. Ook kan de opdrachtgever zelf verzoeken om architecten die men kent in het team op te nemen. Het assessment-team heeft als taak om op het juiste moment de juiste discussies te starten, kritische vragen te stellen, *stakeholders* en architecten uit te dagen en hen een spiegel voor te houden. In het vaststellen van de bevindingen en aanbevelingen wordt gestreefd naar consensus bij alle deelnemers. De organisatie van het assessment is in handen van een facilitator. Deze zorgt ervoor dat de vraagstelling duidelijk wordt gedefinieerd, zoekt hierbij de meest optimale aanpak, selecteert de deelnemers, modereert de workshop en draagt zorg voor een goede rapportage van de resultaten naar de opdrachtgever.

Ervaringen

Architectuur assessments volgens deze methodiek zijn inmiddels uitgevoerd bij verschillende leveranciers van software intensieve, embedded systemen. Vooral Philips blijkt in toenemende mate open te staan voor assessments uitgevoerd door een externe partij. Hoewel de methodiek toepasbaar is voor systeemarchitecturen in het algemeen, ging het bij de meeste assessments om het beoordelen van de softwarearchitectuur. Opdrachtgevers zijn doorgaans software managers, R&D managers of business managers. De assessments worden nog voornamelijk op ad hoc basis uitgevoerd omdat ze nog geen integraal onderdeel uitmaken van de bedrijfs- en ontwikkelprocessen van de opdrachtgever.

De redenen voor het laten uitvoeren van een assessment zijn zeer divers. Zo kan een assessment worden uitgevoerd als onderdeel van een ontwikkelproject. Het doel is dan om de architectuurrisico's in kaart te brengen op een moment waarop de architectuur nog redelijkerwijs aangepast kan worden. Dit was onder meer het geval bij een assessment van het iPronto systeem voor *high-end remote control* van Philips Remote Control Systems in Leuven. In andere gevallen gaat het om een bestaand product en is er behoefte aan duidelijkheid over de mate waarin belangrijke niet-functionele eisen door de architectuur worden ondersteund. Zo werd tijdens een assessment voor de TV-divisie van Philips Consumer Electronics onderzocht in welke mate de architectuur van een herbruikbaar software-platform de gemeenschappelijkheid in de producten uitbaat. Ook zijn er assessments die zich geheel concentreren op het beoordelen van één of enkele belangrijke architectuurkeuzes. Een voorbeeld hiervan is een recent uitgevoerde assessment bij de afdeling Magnetic Resonance van Philips Medical Systems. Daar werd onder meer de keuze voor een *operating system* geëvalueerd. Tot slot komt het voor dat de opdrachtgever zelf een architectuuronderzoek heeft uitgevoerd, en men de resultaten hiervan tijdens een extern assessment wil laten beoordelen. Bij Bosch Security Systems in Breda had het team van architecten een ATAM-sessie uitgevoerd en werden de resultaten daarvan tijdens een architectuur assessment gespiegeld aan het team van externe architecten.

Leerpunten

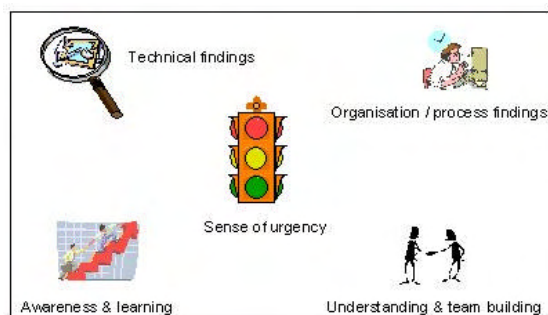
Vanwege het innovatieve karakter van architectuur assessments zijn er de laatste jaren veel ervaringen opgedaan waarmee de aanpak verder is vervolmaakt. Een van de belangrijkste leerpunten is dat het succes van een assessment wordt afgedwongen door een goede voorbereiding. Tijdens deze voorbereiding moeten de verwachtingen van de deelnemers zorgvuldig worden opgelijnd en een vraagstelling worden geformuleerd die geen misverstanden toelaat. Vervolgens moet een aanpak worden ontwikkeld die maximaal aansluit op deze vraagstelling en op de mogelijkheden in termen van budget, doorlooptijd en beschikbaarheid van deelnemers. Architectuur assessments blijken daarom altijd maatwerk te zijn. De standaard aanpak die in eerste instantie is ontwikkeld is dan ook vervangen door een verzameling algemene concepten, aangevuld met richtlijnen die variaties hierop ondersteunen. Zelfs tijdens de plenaire workshop wordt nog maximale flexibiliteit ingebouwd door regelmatig stil te staan bij de aanpak en deze bij te stellen indien hiertoe aanleiding mocht zijn.

Daarnaast is gebleken dat de deelnemers aan het assessment zeer zorgvuldig moeten worden geselecteerd. *Stakeholders* en architecten moeten niet alleen inhoudelijk bij kunnen dragen. Ze moeten ook voldoende open naar elkaar zijn en geen verborgen agenda's hebben. Architectuur assessments vinden altijd plaats in een (meer of minder sterke) politieke context. De selectie van deelnemers is doorgaans het belangrijkste instrument om het assessment te "de-politiceren".

Bij een enkele assessment werden de aanbevelingen onvoldoende omgezet in daadwerkelijke acties ter verbetering van de architectuur. Om dit te voorkomen wordt in de voorbereidingsfase voortaan een "*readiness check*" opgenomen. Daarin wordt nagegaan hoe "klaar" de organisatie is voor een assessment en voor het uitvoeren van effectieve vervolgacties. Wordt aan deze voorwaarden onvoldoende voldaan dan wordt getracht hierin eerst verandering te brengen. Is dit niet mogelijk dan wordt geadviseerd om van het assessment af te zien.

Uitkomsten

Een belangrijke vraag is natuurlijk wat architectuur assessments uiteindelijk opleveren. Voorafgaand aan een assessment is deze vraag vaak moeilijk te beantwoorden. Dat kan alleen als er sprake is van een concreet probleem (zoals te hoge onderhoudskosten van een product) en er aanwijzingen zijn dat de architectuur hiervoor verantwoordelijk is. In dat geval kan vooraf een kosten-baten analyse worden opgesteld. Assessments worden echter ook op preventieve basis uitgevoerd om zeker te stellen dat er geen significante risico's in de architectuur aanwezig zijn. Blijkt dit daadwerkelijk het geval te zijn dan wil dit uiteraard niet zeggen dat het assessment zinloos is geweest.



Figuur 2: Resultaten van een architectuur assessment

De resultaten van een assessment kunnen in vijf categorieën worden onderverdeeld (Figuur 2). Allereerst is er de zogenaamde *sense of urgency*. Dit is het gevoel dat zich tijdens het assessment ontwikkelt over de noodzaak van verbeteringen. Een sterke *sense of urgency* is in grote mate bepalend voor de kracht en het momentum waarmee verbeteringen in gang worden gezet. Deze verbeteringen worden gebaseerd op de technische architectuurbevindingen die uit het assessment komen. Deze worden uitgedrukt in sterke en zwakke punten, risico's en aanbevelingen. Verder blijkt dat vrijwel ieder assessment ook bevindingen oplevert over de kwaliteit van het architectuurproces. Dit is de manier van werken die architecten volgen tijdens de ontwikkeling, het onderhoud en de evolutie van de architectuur. Tot slot zijn er twee categorieën van zogenaamde "neven-producten". Zo verhoogt een assessment de bewustwording van het belang van een goede architectuur en wordt er tijdens de workshop veel geleerd. Tot slot blijkt de samenwerking tussen *stakeholders* en architecten na afloop van het assessment vaak te zijn verbeterd. Dit komt omdat de workshop sterk bevorderend werkt op de teamgeest en op het wederzijds begrip voor elkaars visies, belangen en uitdagingen.

Architectuurbevindingen

Analyseren we de technische architectuurbevindingen dan blijkt er weinig overeenkomst te zijn uit de verschillende assessments. Dit komt in de eerste plaats omdat de vraagstelling bij assessments heel divers is, en daarmee ook de aspecten van de architectuur die worden onderzocht. Bovendien zijn architecturen sterk gekoppeld aan het toepassingsgebied, waardoor bepaalde aspecten van een architectuur in het ene gebied sterker voorkomen dan in het andere. In tabel 1 is desondanks getracht een overzicht te geven van de vijf meest voorkomende categorieën waaruit zwakke punten in de architectuur naar voren zijn gekomen. Daarbij moet worden opgemerkt dat het totaal aantal gevonden categorieën een veelvoud hiervan is.

Categorie	Verklaring
Executie architectuur	De dynamische architectuur in termen van run-time elementen, resource allocatie, timing, scheduling, etc.

Vorbereiding op innovaties	Mate waarin de architectuur is voorbereid op de meest waarschijnlijke productinnovaties.
Ontwerp-erosie	Geleidelijke aantasting van architectuurkwaliteit door onvoldoende aandacht voor goed onderhoud.
Balans SW - HW architectuur	Verdeling van de systeemverantwoordelijkheid over hardware en software, en de afstemming daartussen.
Kwaliteit van de decompositie	Mate waarin de decompositie in componenten is afgestemd op de hoofdverantwoordelijkheden van het systeem.

Tabel 1: De vijf meest voorkomende categorieën waaruit technisch-inhoudelijke bevindingen naar voren komen

Procesbevindingen

Hoewel geen primaire doelstelling van architectuur assessments, leveren ze zoals vermeld ook nuttige informatie op over de kwaliteit van het architectuurproces (Tabel 2). Tijdens de workshop wordt immers veel aandacht besteed aan de zwakke punten en risico's in de architectuur, waardoor ook gemakkelijk discussies worden gestart over de oorzaken hiervan. Wanneer er veel of belangrijke problemen op het procesvlak aanwezig zijn kunnen deze de discussies tijdens de workshop sterk domineren. In tegenstelling tot de technische bevindingen blijkt er op het procesvlak wel veel overeenkomst te zijn uit de verschillende assessments. Dit komt vooral omdat het proces minder sterk door het toepassingsgebied wordt gedictieerd.

Categorie	Verklaring
Transparantie van architectuur	Mate waarin beslissingen op managementniveau worden ondersteund door volledige en betrouwbare architectuurinformatie.
Leiderschap van architecten	Bevordert pro-actieve ontwikkeling en evolutie van de architectuur op basis van visie, langetermijndoelen en architectuur <i>roadmaps</i> .
Definitie van architectuur processen	Het vastleggen, communiceren, trainen en continu verbeteren van de processen.
Samenwerking architecten en <i>stakeholders</i>	Bevordert gemeenschappelijk begrip van niet-functionele eisen, randvoorwaarden, technische mogelijkheden en beperkingen.

Motiveren van architectuurkeuzes	Mate waarin architectuurkeuzes expliciet worden gemotiveerd tegen eisen en randvoorwaarden.
----------------------------------	---

Tabel 2: De vijf meest voorkomende categorieën waaruit procesbevindingen naar voren komen

Vervolgacties

Kijken we naar de maatregelen die op basis van de assessments worden genomen, dan is dit veelal een combinatie van architectuurverbeteringen en procesverbeteringen. In de meeste gevallen werden deze al tijdens of vlak na het assessment opgestart. De verbeteringen van de architectuur kunnen vaak in relatief korte tijd worden uitgevoerd. Verbeteringen van het proces vereisen doorgaans een grotere doorlooptijd, maar hebben als voordeel dat daarmee de kwaliteit van alle toekomstige architecturen beter wordt gewaarborgd.

In enkele gevallen waren de technisch inhoudelijke bevindingen zodanig ernstig dat er ingrijpende maatregelen genomen moesten worden. Doorgaans hebben de resultaten echter niet het karakter van “rampen” maar van “verbeterkansen” (met verschillende niveaus van prioriteit). De architecten zijn zich van een deel van deze verbetermogelijkheden vaak in meer of mindere mate bewust. Het assessment zorgt voor het extra “duwtje” dat nodig is om deze in concrete acties om te zetten door ze expliciet te maken en door het belang ervan bij het management te benadrukken.

Toegevoegde waarde

Uit evaluaties van de uitgevoerde assessments blijkt dat opdrachtgevers de toegevoegde waarde doorgaans als hoog waarderen. Als belangrijkste reden wordt genoemd dat het assessment veel zichtbaarheid (transparantie) in de architectuur geeft die door managers in het algemeen sterk wordt gemist. Men krijgt een goed onderbouwd inzicht in de risico's die in de architectuur zijn opgesloten en een verzameling concrete aanbevelingen. Omdat tijdens de workshop naar consensus wordt gestreefd vinden deze aanbevelingen ook draagkracht bij de architecten zelf. Daarmee wordt aan een belangrijke voorwaarde voor succesvol verbeteren voldaan.

De betrokkenheid van een team van externe architecten die elk hun eigen referentiekader meebrengen werkt in het algemeen sterk verfrissend op de lokale architecten. Doordat bestaande denkkaders worden opengemaakt leert men op andere wijze aan te kijken tegen de grote technische uitdagingen die het vak van architect met zich meebrengt.

De term “neven-producten” blijkt in de praktijk onrecht te doen aan de grote waarde die hieraan door deelnemers en opdrachtgevers wordt toegekend. Het leereffect en de positieve werking op de teamsfeer worden soms als even belangrijk gezien als de beoordeling van de architectuur. De toename van het wederzijds begrip tussen *stakeholders* en architecten blijkt een sterke basis te

bieden voor verbeteringen van de onderlinge samenwerking. In de praktijk blijkt deze samenwerking een van de belangrijkste succesfactoren te zijn voor de ontwikkeling van kwalitatief hoogwaardige systeem- en softwarearchitecturen.

Over de auteur

Herman Postema is werkzaam als Principal Consultant bij LogicaCMG Nederland B.V. Hij houdt zich intensief bezig met prestatieverbetering van product-ontwikkelorganisaties. Herman is betrokken bij de ontwikkeling van methodieken voor architectuur-evaluatie en heeft architectuur assessments gefaciliteerd bij grote leveranciers van embedded systemen.

n Kwaliteit beter op de SPIder ‘kaart’ geplaatst

Vanuit de samenwerkende bedrijven geïnteresseerd om kwaliteit beter op de kaart te krijgen heeft een task force voor de zomer een eerste Business Case opgezet. Een eerste inventarisatie is gemaakt om te bepalen in welk aanbod voorzien zou moeten worden (kennisdeling, seminars/conferenties, vraagbaak, netwerken, et cetera) en welke vraag kan worden voorzien (wie, hoeveel, wat wil men zien/hebben/bereiken, et cetera). Over de vraagkant van de doelgroep is onlangs een enquête uitgezet. We verwachten dat deze maand de resultaten binnen zullen komen. Parallel aan deze actie komt de task force deze maand bij elkaar om de strategie betreffende Q community en Q platform nader uit te werken. Er wordt al voorzichtig gedacht om een Q special in het voorjaar van 2006 te organiseren. Voor Q geïnteresseerden maar ook voor de andere lezers, zou ik willen zeggen “blijf de Koerier goed volgen want daarin wordt verslag gedaan van onze voortgang op het Q gebied.

In de vorige Koerier deed ik een oproep om feedback te verkrijgen van werkgroepen en lezers om mede te bepalen welke kant wij op willen gaan op dit gebied. Tot nu toe is daar matig gevolg aan gegeven. Ik wil de lezer nogmaals van harte uitnodigen om te reageren met meningen, (tegen)stellingen en kopij.

Martin Muller

Martin.Muller@LogicaCMG.com

n De werkgroepen

n Oproep nieuwe werkgroep “Personal SPI”

Via deze oproep willen we inventariseren of er behoefte is aan een werkgroep die zich bezighoudt met SPI op het persoonlijke niveau.

Zaken als SEI's Personal Software Process (PSP) en TSP, maar wellicht ook aanverwante methodieken, technieken en tools kunnen in deze werkgroep de revue passeren. Zou je willen meepraten in deze werkgroep, laat het ons weten.

Je kunt je aanmelden bij Cantrijn, email info@cantrijn.nl.

n Werkgroep "SPI in kleine organisaties"

Wilt u een bijeenkomst van de werkgroep "SPI in kleine organisaties" bijwonen, neem dan contact op met: Herman Rave (voorzitter), tel: 06-53 231 662, hg.rave@chello.nl of Tjeu Naus, tel: 0495-633221, Tjeu.Naus@nbg-industrial.nl

n Werkgroep "Metrieken"

Contactpersoon: Robert van Lieshout E-mail: r.v.lieshout@simbon.nl Tel: +31-499-313575 +31-6-23921989

n Werkgroep "SPI Invoeringsstrategieën"

Contactpersoon: André Heijstek
Tel: 0182-689321 / 06-48476451, e-mail: Andre.Heijstek@ImprovementFocus.com

n Werkgroep "Integrale SPI strategieën"

Werken met modellen, wat doe je ermee en waarvoor zijn ze toepasbaar in een organisatie. De SPIDER werkgroep Integrale SPI-Strategieën is met deze vraagstelling aan de slag gegaan.

Vanaf begin 2003 heeft de werkgroep zich bezig gehouden met diverse case studies over integrale procesverbetering. Integraal betekent in dit verband: ontwikkeling, vrijgave, beheer, wijziging, alignement van business processen met IT-processen, enz. Na deze case studies is de werkgroep gaan werken aan een vergelijkingsmatrix van modellen. Voor het samenstellen van deze vergelijkingsmatrix zijn door de werkgroep de volgende activiteiten uitgevoerd:

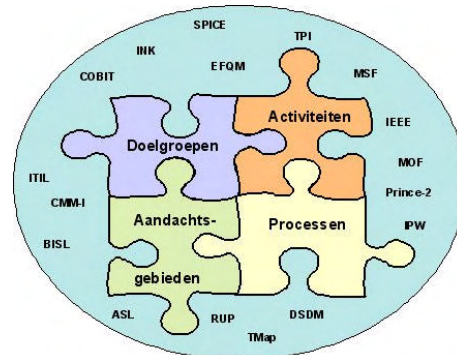
1. Inventariseren modellen.
Allereerst zijn de modellen geïnventariseerd die actueel toegepast worden binnen organisaties. Het betreffen modellen uit een breed werkgebied. Van ontwikkeling tot beheer en van informatiestrategieplanning tot projectmanagement etc.
2. Beschrijven modellen.
Met behulp van een, binnen de werkgroep ontwikkelde, template zijn de geïnventariseerde modellen beschreven.
3. Samenstellen vergelijkingsmatrix.
Voor het samenstellen van een vergelijkingsmatrix zijn toepassingsgebieden vastgesteld. Deze toepassingsgebieden hebben betrekking op de categorieën: doelgroepen; aandachtsgebieden; activiteiten; processen en niveau. Vervolgens is in discussiebijeenkomsten per model vastgesteld op welke toepassingsgebieden het model betrekking heeft.

Het resultaat van onze werkgroep zal op 10 november 2005 in de plenaire SPIDER bijeenkomst worden gepresenteerd. Het resultaat wordt samengevat in een modellenboekje met daarin de volgende onderwerpen:

- Een vergelijkingsmatrix van modellen; hieruit kan afgelezen worden welk model voor welk toepassingsgebied geschikt is. Overeenkomsten en verschillen tussen de

modellen onderling kunnen hierdoor ook uit de tabel afgelezen worden.

- Beschrijving van modellen. Elk model uit de vergelijkingsmatrix wordt beschreven aan de hand van de gedefinieerd aantal kenmerken.



Wij hopen met dit modellenboekje een bijdrage te leveren aan het verbeteren van inzicht in modellen. Modellen groeien in de loop der tijd op basis van ervaringen die ermee opgedaan wordt. Wij willen jullie daarom aanmoedigen om onze kennis en ervaringen met de diverse modellen te delen op 10 november, maar ook om uw ervaringen met ons te delen opdat dit boekje mee kan groeien.

Contactpersoon voor de werkgroep is:

Mario van Os, tel: 06-22516903,
mario.van.os@sogeti.nl

n Nieuwsberichten & evenementenkalender

Evenementen, Conferenties en Seminars

- INCOSE afternoon event
Requirements Engineering
20 Okt 2005 – Utrecht
www.incose.nl
- AGILE seminar
Tweede Agile Seminar Nederland
26 Okt 2005 – Nieuwegein
http://www.agileallianceeurope.org/Nederland/agileseminarautumn2005/document_view
- Bits & Chips Management Event
Het managen van complexe systeemontwikkeling
2 Nov 2005 – Vught
www.bits-chips.nl/events.asp
- Testnet event
Testen met schaarse middelen
2 Nov 2005 – Nieuwegein
www.testnet.org/Produktie/Evenementen/Evenementen.htm
- EuroSPI 2005 Conference
Process Improvement Methodologies and Technologies, Business Strategies, Human Factors, Knowledge and Innovation
9 – 11 Nov 2005 – Budapest, Hungary
www.eurospi.net (SPIDER donateurs krijgen 20% korting!)

- SPIder plenaire sessie
Integrale SPI strategieën
10 Nov 2005 – Utrecht
www.st-spider.nl
- Mechatronica congres
14 Nov 2005 – Eindhoven
www.mechatronica2005.nl
- LaQuSo – symposium
Where innovations in Testing are presented
24 Nov 2005 – Eindhoven
www.laquso.com/VVSS2005.php
- EuroSun 2005 Conference
European Software Managers Update Network
28 Nov – 1 Dec 2005 – Kopenhagen, Denemarken
www.eurosun2005.com (SPIder donateurs krijgen 10% korting!)
- Testnet thema-avond
Common Criteria testen
15 Dec 2005 – Nieuwegein
www.testnet.org/Produktie/Evenementen/Evenementen.htm
- ASL – themabijeenkomst
Zo werk je met ASL
8 Nov 2005 – Soesterberg
daniel.fokker@ordina.nl



n Colofon

De SPIder redactie bestaat uit:
Cees Michielsen en Cantrijn Secretariaten.

Voor reacties en vragen m.b.t. de **SPIder Koerier** kunt u zich wenden tot:

Redactie SPIder Koerier e-mail: Koerier@st-spider.nl

Indien u in de toekomst een herinnerings-bericht wilt ontvangen over de datum van kopijsluiting, stuur dan een e-mail "opname SPIder kopijlijst" naar Koerier@st-spider.nl.

Informatie over SPIder is te vinden op de website: www.st-SPIder.nl.

Voor reacties en bijdragen op de **SPIder website** kunt u zich richten tot:

Redactie SPIder web, Niels Malotaux
E-mail: niels@malotaux.nl

Deze Koerier kwam tot stand met medewerking van

- ITIB (www.itib.net)
- Cantrijn Secretariaten

Calls for papers

- ESEPG 2006 Amsterdam
<http://www.espi.org/sect03.asp>

n Deelname in SPIder

Indien u actief wilt participeren in SPIder en de Koerier in de toekomst wilt ontvangen, kunt u zich aanmelden als deelnemer in SPIder bij:

Secretariaat Stichting SPIder

p/a Cantrijn Secretariaten
Postbus 2047, 4200 BA GORINCHEM
Tel: 0183 - 62 00 66, fax: 0183 - 62 16 01
E-mail: info@st-SPIder.nl.

Aanmelding kan ook via het aanmeldingsformulier op de website van SPIder: www.st-SPIder.nl.